

LAB # 01

Introduction to Visual Studio .NET Integrated Development Environment

Integrated Development Environment (IDE)

An integrated development environment (IDE) is a software suite that consolidates basic tools required to write and test software.

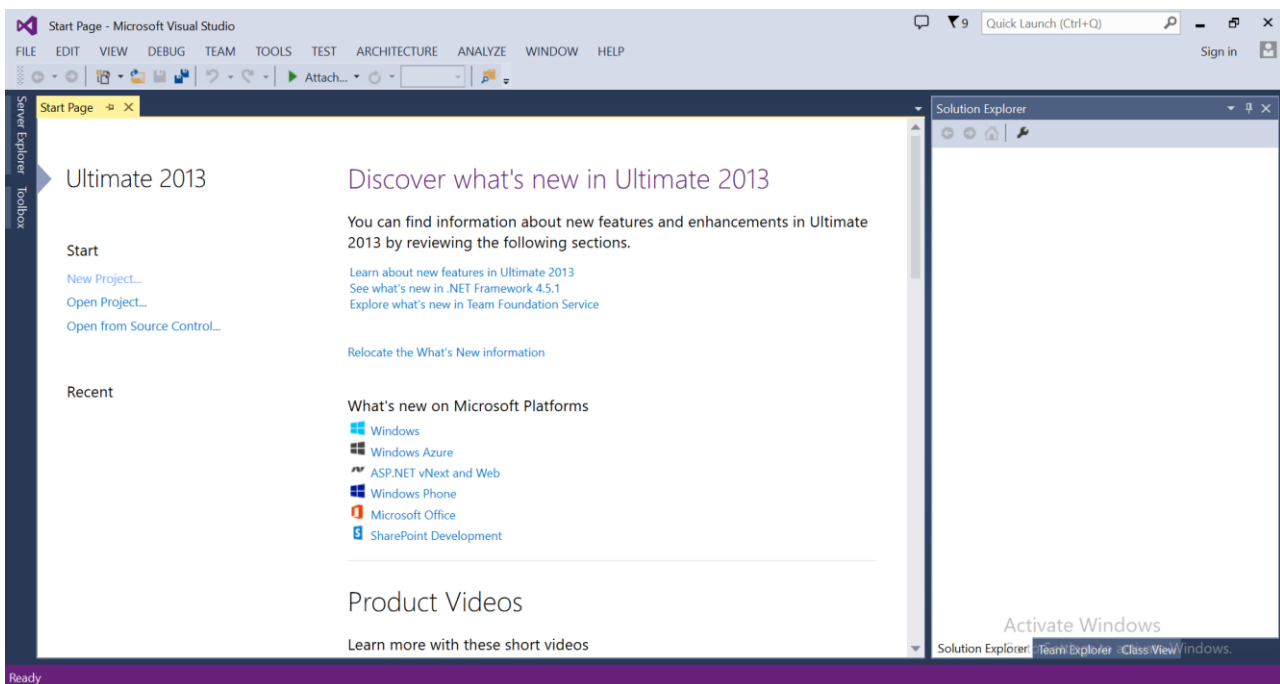
Developers use numerous tools throughout software code creation, building and testing. Development tools often include text editors, code libraries, compilers and test platforms. Without an IDE, a developer must select, deploy, integrate and manage all of these tools separately. An IDE brings many of those development-related tools together as a single framework, application or service. The integrated toolset is designed to simplify software development and can identify and minimize coding mistakes and typos.

Program Development with Microsoft Visual C++ 2010

This lab session introduces the **Integrated Development Environment (IDE)** of Microsoft Visual C++ 2010 and shows how to enter, edit, save, retrieve, compile, link, and run a C++ program in such an environment.

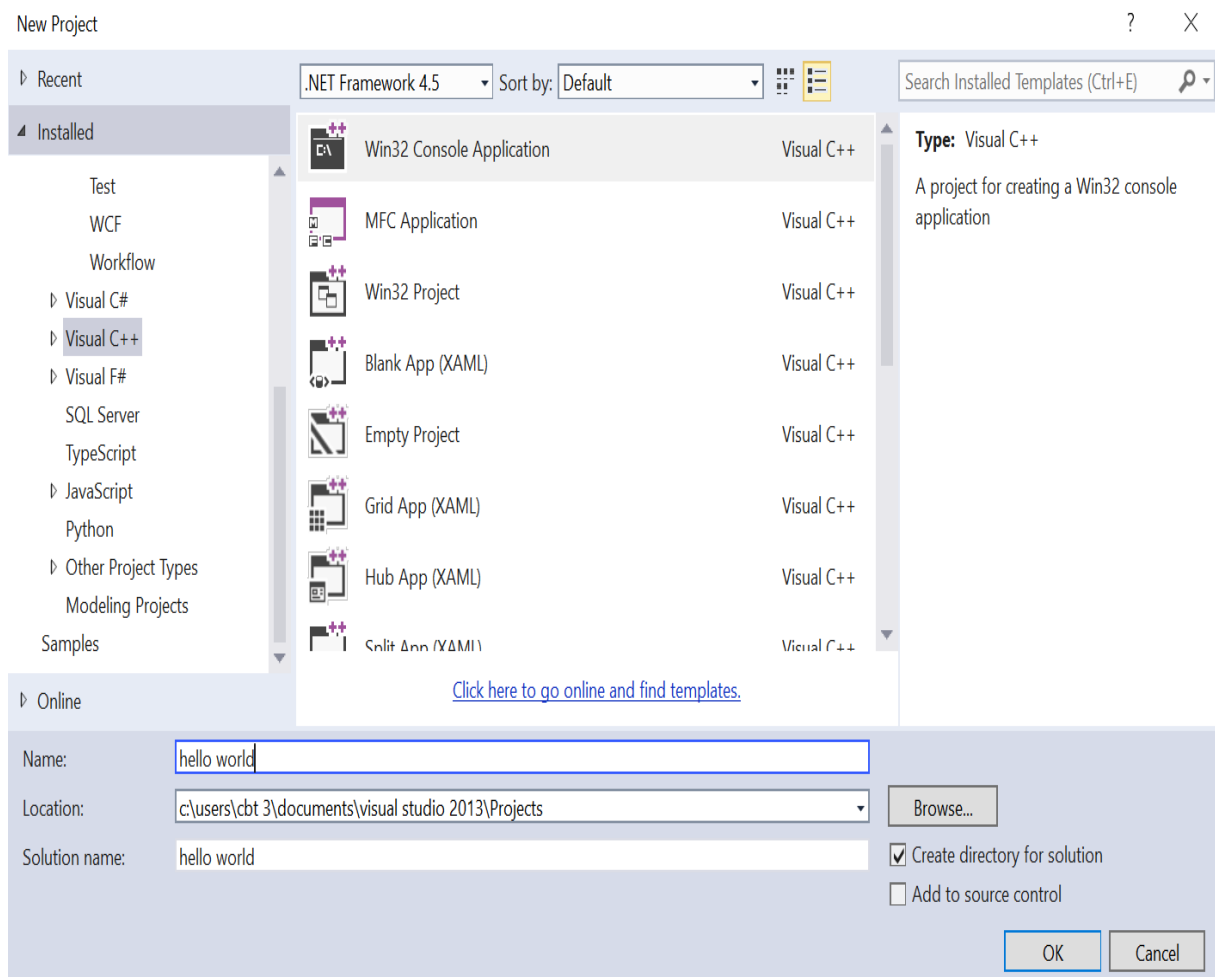
Single-file Project: The *Hello World* Program

Step 1: Launch the MS Visual C++ 2013 software from task bar. The main window of Visual Studio 2010 should be similar to the below display:



- Hereafter, all system defined terms including menu items such as File will appear in bold and all entries made by programmers such as a filename are italicized.
- If the Solution Explorer window on the left is not shown, click View in the menu bar and then click Solution Explorer to display it.

Step 2: In the menu bar, click **File → New → Projects...** to display the **New Project** dialog box shown below. In the **New** dialog box shown below, select by clicking **Visual C++** in the **Installed Template** pane and **Win32 Console Application** in the middle pane. Then enter a name for the project (e.g., *hello world* as shown) in the Name box, select the location or folder to store project files (e.g., C:\Users\.... as shown) by clicking the **Browse...** Note that there is no need to enter a name in the Solution name box; the system fills the project name in it by default.



Click on the **OK** button to display the **Win32 Application Wizard – *hello world*** window shown below:

Win32 Application Wizard - hello world

? ×

**Welcome to the Win32 Application Wizard**

Overview

Application Settings

These are the current project settings:

- Console application

Click **Finish** from any window to accept the current settings.

After you create the project, see the project's readme.txt file for information about the project features and files that are generated.

< Previous

Next >

Finish

Cancel

Click the **Next** button to display the following dialog box:

Win32 Application Wizard - hello world

? ×

**Application Settings**

Overview

Application Settings

Application type:

- ☐ Windows application
- ☒ Console application
- ☐ DLL
- ☐ Static library

Additional options:

- ☒ Empty project
- ☐ Export symbols
- ☒ Precompiled header
- ☒ Security Development Lifecycle (SDL) checks

Add common header files for:

- ☐ ATL
- ☐ MFC

< Previous

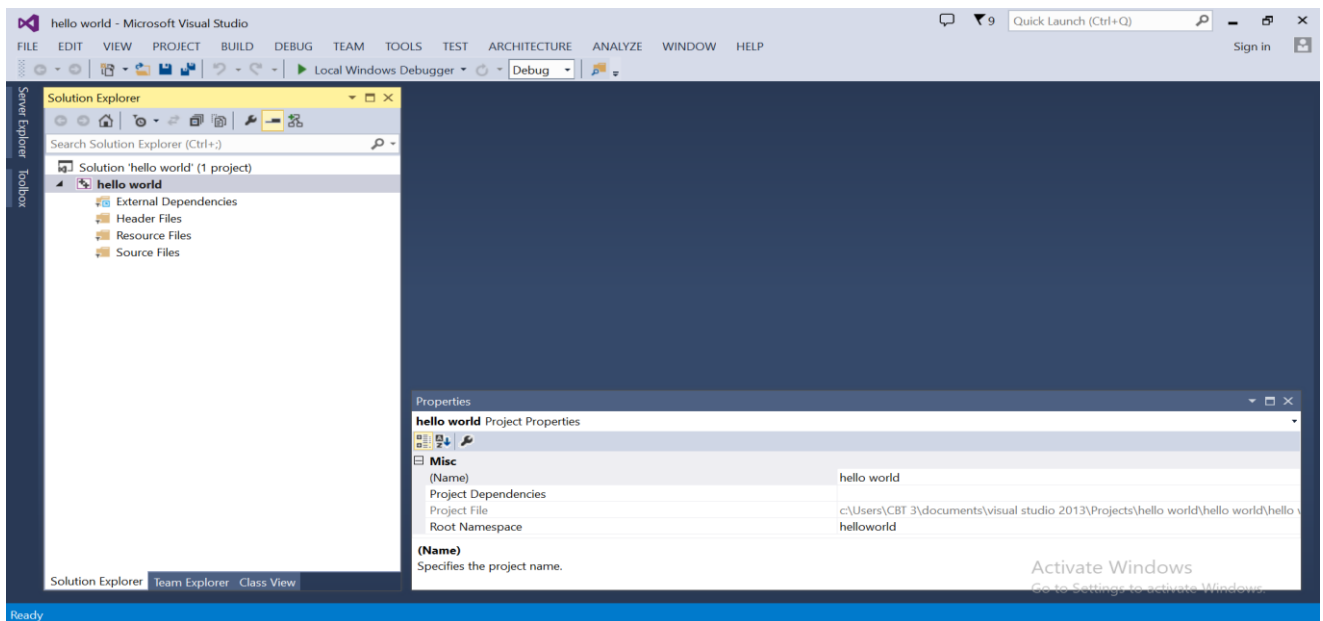
Next >

Finish

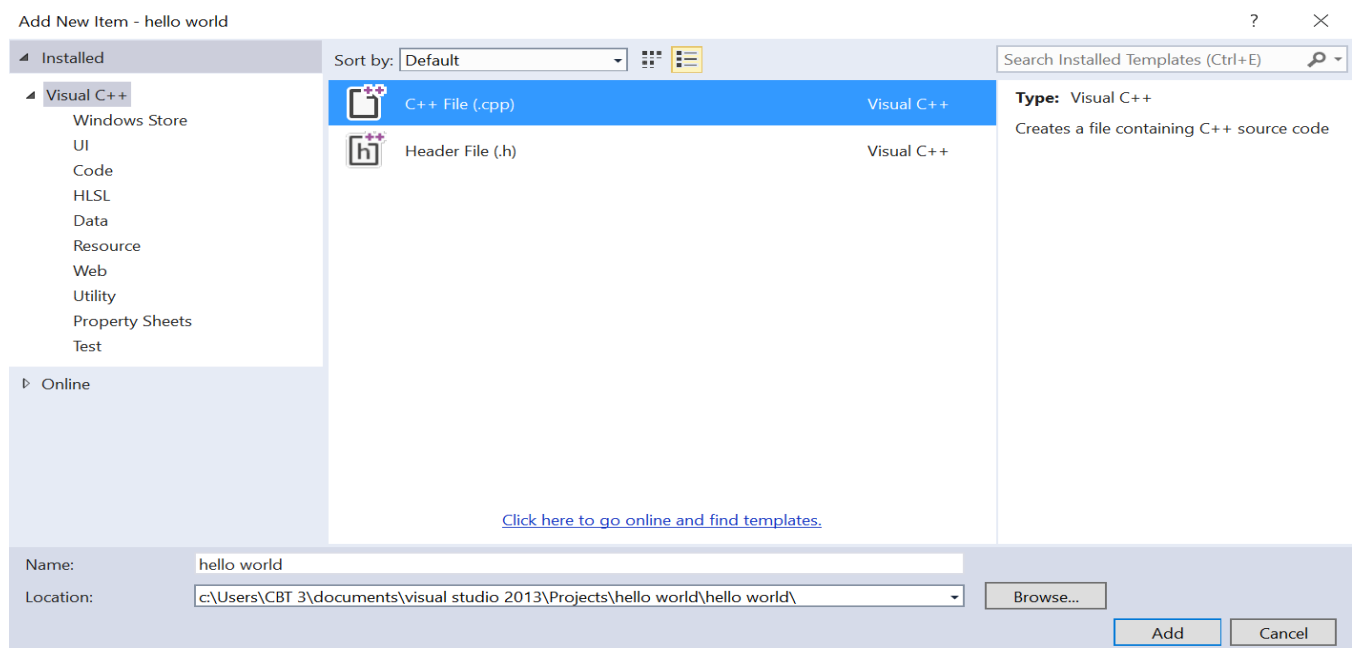
Cancel

Check the **Empty project** box as shown above and click on the **Finish** button to proceed to the next step.

Step 3: Now the system displays the following window.

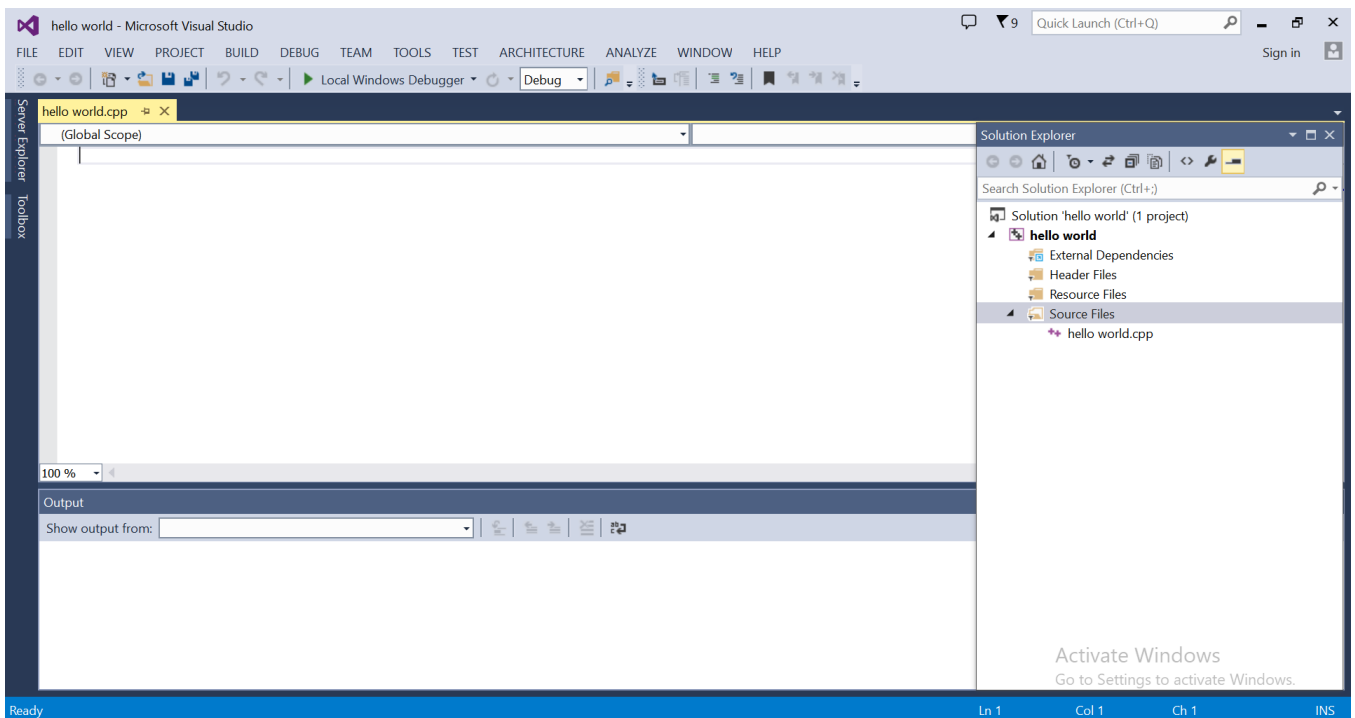


Right click on the **Source Files** folder in the **Solution Explorer** pane. In the popup menu, click **Add** then **New Item...** to display the following **Add New Item – hello world** dialog box:



Select **C++ Files (.cpp)** by clicking on it in the middle pane and enter an arbitrary file name (e.g., *hello world*). Click **Add** to proceed to the next step.

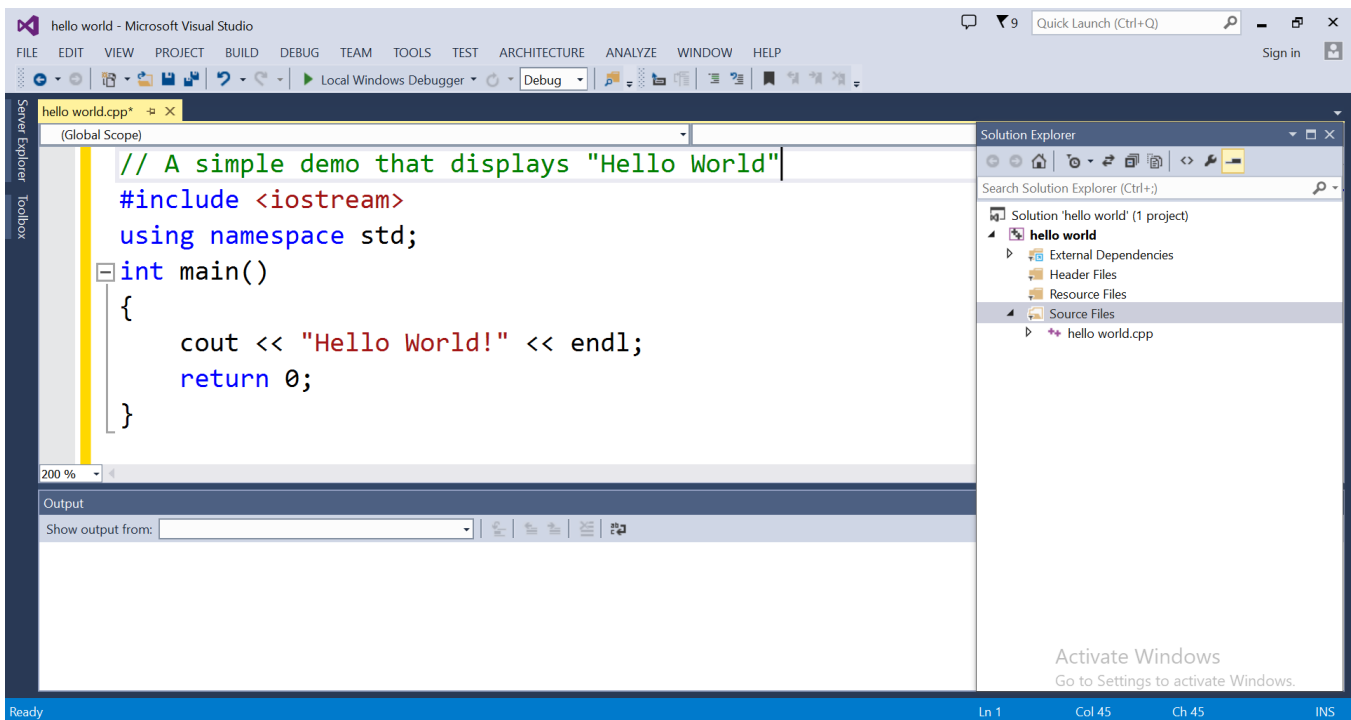
Step 4: The system displays the below window. The **Source Folder** in **Solution Explorer** pane contains the *hello world.cpp* file that was just added. The blank editing area/board is displayed with a *hello world.cpp* tab to enter the C++ source code.

**Source Code:**

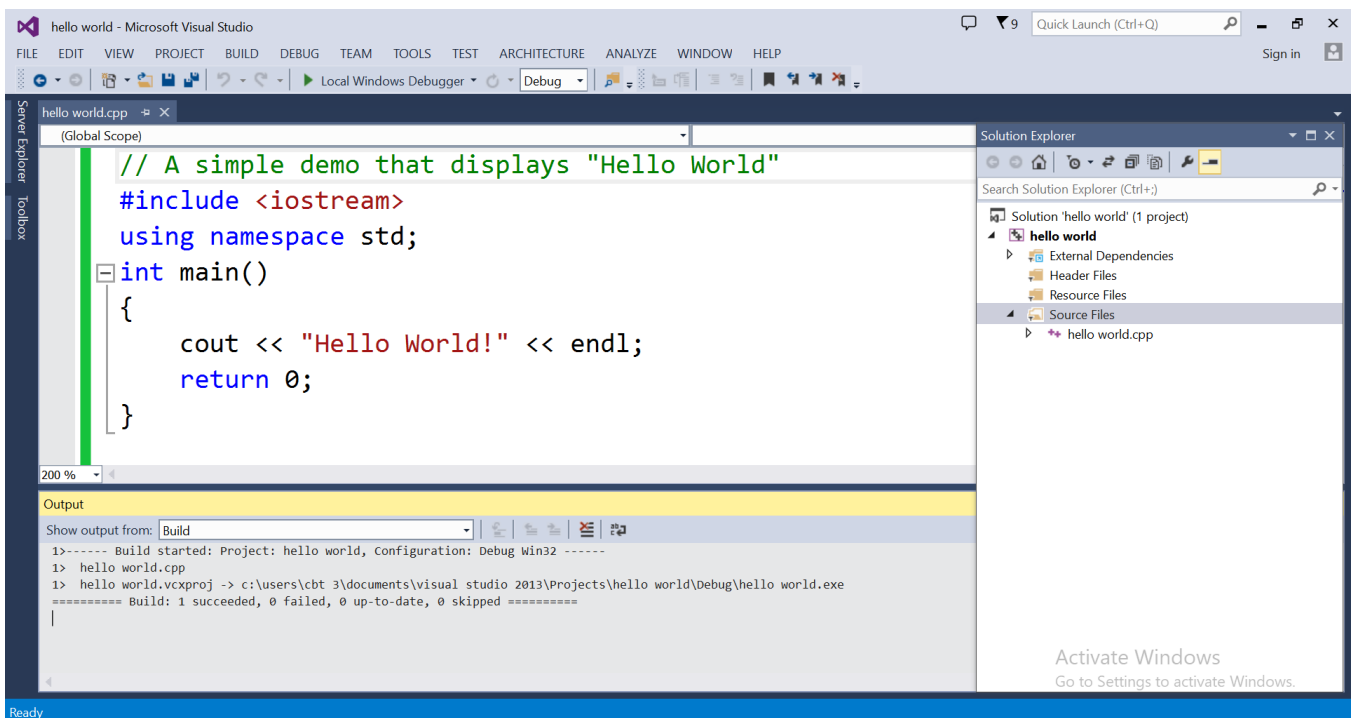
```
#include <iostream>
using namespace std;
int main()
{
    cout << "Hello World!" << endl;
    return 0;
}
```

- **#include:** Lines beginning with a hash sign (#) are directives for the preprocessor. The directive `#include` tells the preprocessor to include the `iostream` standard file.
- **using namespace std:** All the elements of the standard C++ library are declared within what is called a namespace, the namespace with the name `std`.
- **int main ():** This line corresponds to the beginning of the definition of the main function. The main function is the point by where all C++ programs start their execution. The word `main` is followed in the code by a pair of parentheses (`()`).
- **cout << "Hello World!":** This line is a C++ statement. `cout` represents the standard output stream in C++, and the meaning of the entire statement is to insert a sequence of characters into the standard output stream.
- **return 0:** The return statement causes the main function to finish. `return` may be followed by a return code. A return code of 0 for the main function is generally interpreted as the program worked as expected without any errors during its execution.

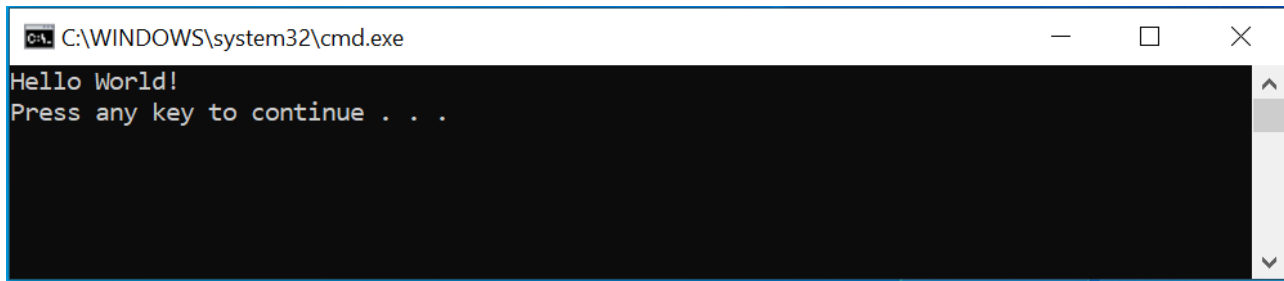
The same window after the source code is entered:



Step 5: To compile, link, load, and execute a program in a single step, click Debug in the menu bar and then click Start Without Debugging. If there is no error in your project, the below message ===== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped ===== is displayed in the Output window as shown below.



Also displayed in a separate window (C:\Windows\system32\cmd.exe) is the program output:

A screenshot of a Windows command prompt window. The title bar shows the path 'C:\WINDOWS\system32\cmd.exe'. The window has standard Windows window controls (minimize, maximize, close). The command prompt shows the output of a program: 'Hello World!' followed by 'Press any key to continue . . .'. The text is displayed in a light color on a dark background.

Lab Task

- 1- Use MS Visual C++ 2013 to create a project named, *Lab1* and then add a .CPP file named, *First* to this project. Now add given piece of code to file *Lab1* and compile this file. Execute this file afterwards and check the output of your program.

```
#include<iostream>
using namespace std;
int main ( )
{
    cout<< " My first Program in C++" ; return 0;
}
```

- 2- Use MS Visual C++ 2013 to create a project named, *Prog2* and then add a .CPP file named, *Second* to this project. Now add given piece of code to file *Prog2* and compile this file. Execute this file afterwards and check the output of your program.

```
#include<iostream>
using namespace std;
int main ( )
{
    int x; x=5;
    cout<< " x = " << x; return 0;
}
```

- 3- What is the output from the following C++ code fragment?

```
#include
<iostream> using
namespace std;
int main(void)
{
    cout << " \n Sir Syed University of Engineering and Technology "
    <<endl;cout << " \t \n programming fundamentals using C++ "
```

```
<<endl;  
return 0;  
}
```

4- What is the output from the following C++ code fragment?

```
#include <iostream>  
using namespace std;  
int main(void)  
{  
    cout << "x =" <<5<<endl;  
    cout << "Y= " <<10<<endl;  
    cout << "x+y= " <<5+10<<endl;  
    return 0;  
}
```
